Como vimos no capítulo anterior, uma linguagem de programação é uma técnica de notação para programar, com a intenção de servir de veículo para:

a) Expressão de um raciocínio algoritmo.

b) Execução automática de um algoritmo no computador.

Na maioria das implementações das linguagens de uso generalizado (FORTRAN, PL/I, COBOL), o segundo item teve um papel preponderante: construções lingüísticas cuja implementação seja menos eficiente são evitadas, quando não no desenho da linguagem, pelo próprio programador ao constatar uma ineficiência. Isso pode criar uma autocensura que faz com que o programador evite essas construções mesmo ao fazer programas abstratos, criando uma espécie extremamente prejudicial de bloqueio mental. Como quase todos sabem, a liberdade de expressão é essencial para o desenvolvimento da criatividade em praticamente todas as atividades humanas. Segundo Wirth, "nós precisamos reconhecer a forte e inegável influência que nossa linguagem exerce em nossos caminhos de pensamento, e que de fato define e delimita o espaço abstrato no qual nós formulamos — damos forma aos — nossos pensamentos".

Diante disso o leitor já pode estar pensando em usar diretamente o português para expressar seus algoritmos. No entanto, esta é uma compreensão incompleta do problema. O uso de uma técnica formal para notação de algoritmos proporciona diversas vantagens, das quais gostaríamos de salientar duas:

— Vimos que somente devemos aceitar um texto como sendo um algoritmo se ele vier acompanhado de uma argumentação convincente. Ora, uma técnica formal de notação nos permite dar esta argumentação de uma maneira razoavelmente concisa.

— Uma técnica formal afasta a possibilidade de ambigüidade de tal maneira que, dado um algoritmo descrito segundo esta técnica e um estado inicial, a sua execução nos leva sempre ao mesmo estado final, através dos mesmos caminhos. "... Linguagens naturais, não formalizadas como são, derivam, tanto a sua fraqueza como o seu poder, de sua imprecisão e seu caráter vago."

Estamos aparentemente diante de um impasse: por um lado, precisamos de liberdade de expressão para formular nossos raciocínios e, por outro, necessitamos de uma notação formal pelos motivos que acabamos de expor.

A solução está na escolha de estruturas adequadas, bem projetadas, com efeitos bem definidos, e que, além disso, não restrinjam a criatividade do programador.

Neste capítulo será utilizada uma pseudolinguagem de programação: o *PORTUGOL*. O que pretendemos então com o PORTUGOL é obter uma notação para algoritmos, a ser utilizada na definição, na criação, no desenvolvimento e na documentação de um programa, e não criar mais uma linguagem de programação. Na definição do PORTUGOL que se segue, procuramos introduzir o mínimo de blocos básicos (de dados e de controle) necessários para a descrição dos algoritmos do curso. O leitor deve ficar ciente de que esta definição satisfez as necessidades deste curso, podendo se sentir à vontade para introduzir novas estruturas, novas construções que melhor se adaptem a seus próprios problemas.

O PORTUGOL é uma pseudolinguagem de programação (simbiose do Português com o ALGOL e PASCAL). A idéia é permitir que com um conjunto básico de primitivas seja possível ao projetista pensar no *problema* e não na *máquina* que vai executar o algoritmo e, por outro lado, não fique muito distante desta mesma máquina. Em outra perspectiva, que o projetista possa pensar na solução do problema e que esta solução seja facilmente implementada no computador.

2.1 INTRODUÇÃO

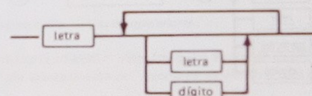
Em toda linguagem, as frases construídas envolvem dois aspectos: a *sintaxe* e a *semântica*. A sintaxe tem a ver com a forma e a semântica com o conteúdo. Considerando o português como linguagem, tomemos uma frase sintaticamente correta (tem verbo, sujeito e objeto, e as palavras estão escritas corretamente):

"Aqui vendem-se frangos abatidos."

A semântica correta desta frase é indicativa de que naquele local existe uma venda de frangos já mortos, e não frangos "deprimidos" ou "anêmicos".

Considerando o PORTUGOL, a sintaxe é *definida* e a forma apresentada deve ser aceita e respeitada como padrão. Para cada declaração e/ou comando a semântica será devidamente explicada.

Como exemplo, vamos definir um elemento básico da linguagem que é o *identificador*. A sua sintaxe é definida pelo diagrama a seguir:



Qualquer caminho que possa ser seguido no diagrama levará a um identificador válido (considerando [letra] como as 23 letras do alfabeto e [dígito] os 10 dígitos do sistema decimal). São, portanto, identificadores sintaticamente válidos:

A, B1, BC3D, A4, Z1976,
ZZZZZ, I, J, SOMA, CONTADOR, ETC.

Um exemplo de identificador inválido seria 2AB ou qualquer outro iniciado por um dígito.

2.2 DECLARAÇÃO DE VARIÁVEIS — TIPOS BÁSICOS

No PORTUGOL, temos quatro tipos básicos, isto é, tipos básicos de dados que podem ser utilizados: